2006.5.18 Okada

実験III　　計算機　　第３回

第３回　数値積分法

いくつかの代表的な数値積分法について学び、実際の問題を解くためのプログラムを作成する。

１．台形近似とシンプソン則

図１－１のような関数 $f(x)$ **の面積を求める問題を考える。**

$$S = \int_a^b f(x)\,dx \tag{1-1}$$

被積分関数 $f(x)$ **が数値だけで与えられていたり、関数が複雑で解析的に積分を求められない場合は数値的に積分を求めるしかない。その近似値を得る方法はいくつも考案されているが、最も簡単な方法は区間を等間隔** h **に区切り、長方形の面積の和として求める方法である。すなわち**

$$S_n = \sum_{k=0}^{n-1} f(x_k)h = \frac{(b-a)}{n}\sum_{k=0}^{n-1} f(x_k), \quad x_k = x_{k-1} + h \qquad . \tag{1-2}$$

ここで、 $f(x_k)$ **は区間の初めで計算するとした。**

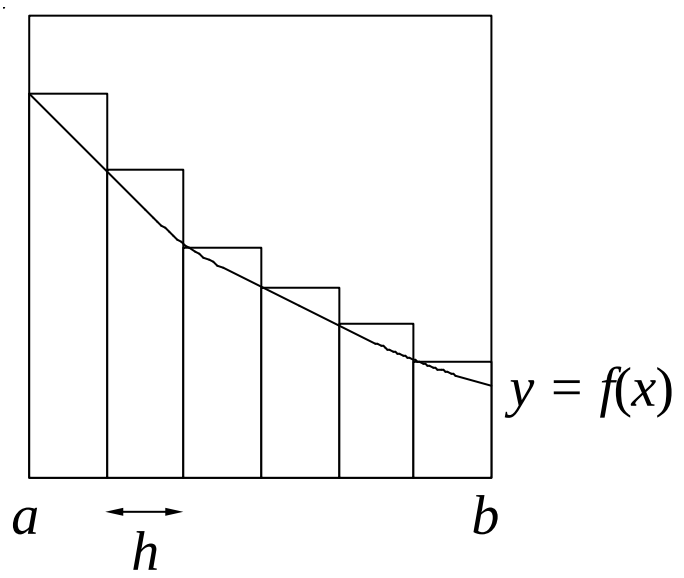

図１－１．長方形近似と台形近似

近似の精度を高めるには、長方形ではなく台形近似を用いる。これは区間の初めの点と終りの点を直線で結び、その台形の面積の総和を求める方法であり、

$$S_n = \left[\frac{1}{2}f(x_0) + \sum_{n=1}^{n-1} f(x_i) + \frac{1}{2}f(x_n)\right]h \tag{1-3}$$

のように計算できる。例えば台形公式(1-3)式によって定積分

$$S = \int_0^1 xe^x \, dx$$

を計算するプログラムは次のようにかける。

```
Public Function fe1#(x#)    'Function プロシージャ（倍精度）
   fe1 = x * Exp(x)              '関数の定義
End Function

Private Sub cmdTRPZ_Click() 'コマンドボタンの名前 cmdTRPZ
Dim a#, b#, s#, n&, x#                  '変数を宣言（倍精度、長整数型）

a = 0: b = 1#: n = 2              '積分区間，分割数
n = 8                             '分割数
h = (b - a) / n                         '刻み幅
x = a                                   'x=a からスタート
    For i = 1 To n - 1
        x = x + (b - a) / n             '積分のための分点
        s = s + fe1(x)                  '式(1-3)の和の部分
     Next i
   s = h * (s + 0.5 * (fe1(a) + fe1(b)))  台形公式

Mains.Print "s="; s   '値を表示

End Sub
```

また、一定の精度を持った面積を計算するためのプログラムは次のように書ける。

```
Private Sub cmdTRPZ_Click() 'コマンドボタンの名前 cmdTRPZ
Dim a#, b#, s#, n&, x#                  '変数を宣言（倍精度、長整数型）

a = 0: b = 1#: n = 2              '積分区間，分割数
sold = 0                                '比較のための面積の初期値を 0 とする

For k = 1 To 20
 n = 2 * n                        '分割数
 h = (b - a) / n                         '刻み幅
 x = a                                   'x=a からスタート
    For i = 1 To n - 1
        x = x + (b - a) / n             '積分のための分点
        s = s + fe1(x)                  '式(1-3)の和の部分
     Next i
   s = h * (s + 0.5 * (fe1(a) + fe1(b)))  台形公式の最終結果

     '古い面積 sold と新しい面積の比較。相対誤差が 10^-6 以下なら結果を出力して、Sub を終了。
     If Abs(s - sold)/s < 10 ^ -6 Then Mains.Print "s="; s: Exit Sub
 sold = s                         ' sold の入れ換え
 s = 0                            ' s を０に戻す（初期化）
Next k

Mains.Print "収束せず"; "s="; s          '収束しない場合、メッセージ表示して終了。

End Sub
```

台形公式よりもさらに精度を上げるためには、隣接する 3 点を 2 次式で結んで面積を求めればよい。こ

れがシンプソンによる積分法である。隣接する 3 点の座標を$(x_0, y_0), (x_1, y_1), (x_2, y_2)$とすると、この 3 点を通る放物線は

$$y(x) = y_0 \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + y_1 \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} + y_2 \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} \qquad (1\text{-}4)$$

で与えられる。ここで$y_i = f(x_i)$である。(x_0, y_0)から(x_2, y_2)までの面積は解析的に

$$S_0 = \frac{h}{3}\left(y_0 + 4y_1 + y_2\right), \quad h = x_1 - x_0 = x_2 - x_1 \qquad (1\text{-}5)$$

と求められる。面積の総和を求めると

$$S_{2n} = \frac{h}{3}\left[f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \cdots + 2f(x_{2n-2}) + 4f(x_{2n-1}) + f(x_{2n})\right] \qquad (1\text{-}6)$$

となる。これは閉じた**拡張シンプソン則**と呼ばれている。**分割数が偶数（$2n$）でなければならない点に注意すること。**

積分区間$[a, b]$の両端での関数の値$f(a)$，$f(b)$を分点x_iに含めた公式を「閉じた積分公式」とよぶ。一方、積分区間の両端または片方の点で関数の計算が困難な場合がある。このような関数では、aとbの間の分点x_iだけを用いて積分を求めなければならない。このような場合には「開いた積分公式」を用いる（開いた積分公式については参考図書[1], [2]などを参照）。

（課題１）拡張シンプソン則(1-6)式を用いて、定積分

$$S_n = \int_0^1 f(x)\,dx, \quad f(x) = 4\sqrt{1-x^2} \qquad (1\text{-}7)$$

を計算するプログラムを以下の手順に従って作成せよ。

(a) **コマンドボタン** Simpson1 **を用意し、$n = 8$として拡張シンプソン則**(1-6)**式を計算し、「**Mains**」フォーム上に結果を表示するプログラムを作成せよ。関数$f(x)$は** Function **プロシージャで定義すること。関数$f(x)$の内容を変更するだけで別の関数の積分を求められるようにするためである。**
（ヒント）偶数項、奇数項の係数がそれぞれ$4f(x_{2i-1})$，$2f(x_{2i-2})$と異なるので、和を別々に計算して最後に足し合わせるか、あるいは Mod 関数（ヘルプ参照）などを利用して偶数項、奇数項の和を同時にとってゆけばよい。いずれの場合も積分区間の端点の関数値$f(a)$，$f(b)$は最後に別途加えればよい。

(b) **コマンドボタン** Simpson2 **を用意する。実行するとS_nが約５桁の精度で収束するまで自動的に$n \to 2n$ $(h \to h/2)$として計算を繰り返し行うプログラムを作れ。「**Mains**」には分割数nと収束した積分値のみを表示させること。また、得られた結果をノートに記録しておくこと。**
（ヒント１）S_nに対応する変数を倍精度実数(例えば Dim s#)で宣言すること。
（ヒント２）S_nの収束は$|S_{2n} - S_n| / S_{2n} < 10^{-6}$で判定して、収束したらループを抜けて$S_n$をフォームに印刷する。なお定積分(1-7)の正確な値はπ=3.141…= 4*Atn(1)である。

(c) **次の積分を５桁の精度で求めよ。コメント文を用いて** Function **プロシージャの内容を変更し、** Simpson2 **を実行すればよい。**
(i) $S = \int_5^8 \frac{1}{x}\,dx$,　　(ii) $S = \int_0^1 \frac{1}{1+x^2}\,dx$

２．ガウスの求積法

２－１．はじめに

Simpson則では積分を求めるための分点は等間隔であったが、分点をうまく配置することによって効率的に数値積分を行う方法もある。その一つにガウス型の積分公式がある。その原理については後ほど詳述するが、基本的には以下の手順にしたがっている。

⑴被積分関数について積分区間上の直交多項式の零点で関数値を求める。
⑵求めた関数値を結ぶ多項式を近似的に求め、その多項式の積分を求める。

という方法である。ガウス型の積分公式は以下のように積分区間によって用いられる公式が異なる。

(a) ルジャンドル・ガウスの積分公式、チェビシェフ・ガウスの積分公式（有限区間）
(b) エルミート・ガウスの積分公式（無限区間）
(c) ラゲール・ガウスの積分公式（半無限区間）

それぞれの名称は分点に利用する直交多項式の名称に由来している。原理は難しそうに思われるかもしれないが、実際の計算は非常に簡単である。公式によって指定された特定の分点で関数 $f(x_i)$ を計算し、ある決まった係数を掛けて、和をとっていくだけである。数式で書くと、

$$S=\int_a^b f(x)\,dx \cong \sum_{i=1}^{N} c_i\, f(x_i) \qquad (2\text{-}1)$$

という形をしている。ここで、分点の位置x_iと係数c_iは公式によって与えられている既知の数値である。ガウス型の公式ではn点の計算公式で$2n-1$次の精度（被積分関数が$2n-1$次までの多項式のとき、厳密な計算結果と一致するということ）をもつ。

２－２．ガウス型の積分公式

２－２－１．直交多項式

ガウス型の積分公式では直交多項式の零点を積分の分点として採用している。ここで、直交多項式系とは以下のように定義される。

区間(a, b)について、n次式$P_n(x)$が、

$$\int_a^b w(x)P_n(x)P_m(x)\,dx=0,\ \ n\neq m\quad (m=0,1,\cdots,n-1),\ \ w(x)\text{：重み関数} \qquad (2\text{-}2)$$

を満たすとき、多項式の列$P_0(x), P_1(x),\cdots, P_n(x),\cdots$は直交多項式系を構成するといい、それぞれを直交多項式と呼ぶ。

直交多項式の零点は全て実数であり、区間(a, b)内にある。例えば、実験２「BASIC入門」で扱ったルジャンドル多項式も直交多項式であり、次の漸化式によって求められる。

$$P_0=1,\ \ P_1(x)=x,\ \ P_n(x)=\frac{2n-1}{n}xP_{n-1}(x)-\frac{n-1}{n}P_{n-2}(x)\ \ (n\geq 2),\ \ -1\leq x\leq 1. \qquad (2\text{-}3)$$

P_nはn次多項式を与える。具体的には

$$P_0=1,\ P_1(x)=x,\ P_2(x)=\frac{3}{2}x^2-\frac{1}{2},\cdots$$

という多項式である。定義によるとルジャンドル多項式は$(a, b)=(-1, +1),\ \ w(x)=1$についての直交多項式である(図２－１)。

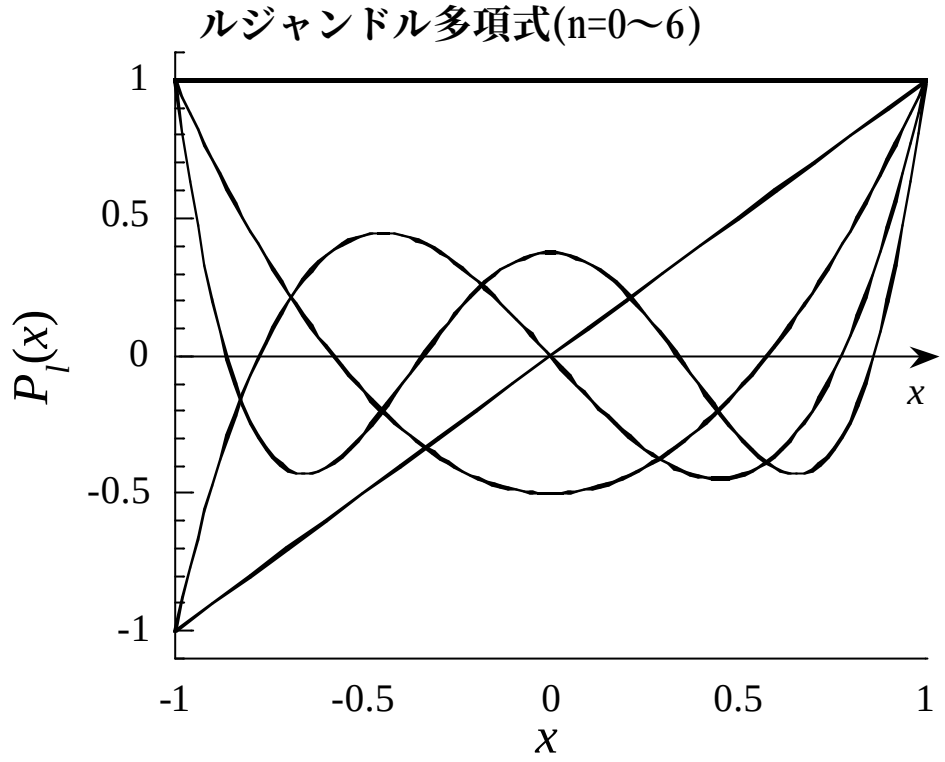

図２－1.ルジャンドル多項式のグラフ。

数値積分を行う上で重要なルジャンドル多項式の性質は、定義域$(-1,1)$の区間における任意の関数を、ルジャンドル多項式の組合せ（線形結合）によって再現できることである。被積分関数をルジャンドル多項式の和によって表し、計算精度を損なうことなく積分を簡単化する方法が実現できる。

２－２－２．ルジャンドル・ガウスの積分公式

求めたい積分が以下のように表されているとする。

$$S=\int_a^b f(x)\,dx \qquad (2\text{-}4)$$

まず、変数変換$x=(a+b)/2+(b-a)\,t/2$によって、ルジャンドル多項式の定義域$(-1,1)$に変換すし、積分を以下のように表す。

$$S=\int_{-1}^{1} f(t)\,dt \qquad (2\text{-}4)'$$

被積分関数$f(t)$が$2n-1$次の多項式を使って表せると仮定する。$f(t)$をn次のルジャンドル多項式$P_n(t)$で割った結果得られた関数を$Q(t)$、余りの関数を$R(t)$とすれば

$$f(t)=P_n(t)Q(t)+R(t) \qquad (2\text{-}5)$$

とかける。したがって積分は

$$\int_{-1}^{1} f(t)\,dt=\int_{-1}^{1} P_n(t)Q(t)\,dt+\int_{-1}^{1} R(t)\,dt \qquad (2\text{-}6)$$

となる。$R(t)$は$n-1$次以下の多項式である。ルジャンドル・ガウス積分公式では、(2-6)の近似値を求める方法として、次のような計算方法を用いている。

⑴ 台形公式やシンプソンの公式などと同様に、関数の横軸をある間隔（分点）で区切って積分値を求めるところは同じである。ただし、その分点は等間隔ではなく、ルジャンドル多項式の零点t_iにとる（零点とは図 2－1 において関数と x 軸が交わる点である）。そうすると、(2-6)式の右辺第一項は 0 になる。なぜなら、

$$\int_{-1}^{1} P_n(t)Q(t)\,dt \cong \sum_{i=1}^{n} c_i P_n(t_i)Q(t) = 0 \quad (\because Pn(t_i) = 0) \tag{2-7}$$

のように計算されるからである（ここで c_i は重み係数であり、台形公式では $h = (b-a) / n$ に相当する）。従って、積分を零点t_iでの関数値を使った和として求めることを前提とするなら、

$$S = \int_{-1}^{1} f(t)\,dt \cong \int_{-1}^{1} R(t)\,dt \quad \rightarrow \quad f(t_i) = R(t_i) \tag{2-8}$$

が成立つことになる。

⑵ $R(t)$ は $(t_i, R(t_i)) = (t_i, f(t_i))$ を通る $n-1$ 次多項式である。n 個の点を通る多項式は、$n-1$ 次のラグランジュ補間多項式とよばれるもので正確に表せることが知られている。すなわち、

$$R(t) = \sum_{i=1}^{n} R(t_i)L_i(t) = \sum_{i=1}^{n} f(t_i)L_i(t) = f(t_1)L_1(t) + f(t_2)L_2(t) + \cdots + f(t_n)L_n(t), \tag{2-9}$$

$$L_i(t) = \frac{(t-t_0)(t-t_1)\cdots(t-t_{n-1})}{(t_i-t_0)(t_i-t_1)\cdots(t_i-t_{i-1})(t_i-t_{i+1})\cdots(t_i-t_{n-1})} \tag{2-10}$$

と表せる。従って、積分(2-8)式は

$$S = \int_{-1}^{1} f(t)\,dt \cong \sum_{i=0}^{n-1} c_i f(t_i), \tag{2-11}$$

重み係数：$c_i = \int_{-1}^{1} L_i(t)\,dt$ (2-12)

と求められる。係数c_iは直交多項式の零点x_iがわかれば計算することができる。分点x_iと重み係数c_iは通常区間(–1, 1)の積分に直されて、標準化された数表となって与えられている（表２－１）。積分区間が有限であればこの公式がよく利用される。ただし、分点x_i、重み係数 c_i は表２－１の値を利用する。分点の数nは被積分関数の形が何次の多項式で近似できそうであるかを見極めてから決める。他の分点の係数や無限区間や半無限区間での積分で利用されるガウス型の積分公式については参考図書を参照すること。

n = 2	
x_i	c_i
-0.5773502692	1.0000000000
0.5773502692	1.0000000000
n = 3	
x_i	c_i
-0.774596669	0.555555556
0.00E+00	0.888888889
0.774596669	0.555555556
n = 4	
x_i	c_i
-0.861136312	0.347854845
-0.339981044	0.652145155
0.339981044	0.652145155
0.861136312	0.347854845
n = 8	
x_i	c_i
-0.960289856	0.101228536
-0.796666477	0.222381034
-0.52553241	0.313706646
-0.183434642	0.362683783
0.183434642	0.362683783
0.52553241	0.313706646
0.796666477	0.222381034
0.960289856	0.101228536

n = 16	
x_i	c_i
-0.989400935	0.027152459
-0.944575023	0.062253524
-0.865631202	0.095158512
-0.755404408	0.124628971
-0.617876244	0.149595989
-0.458016778	0.169156519
-0.281603551	0.182603415
-0.09501251	0.18945061
0.09501251	0.18945061
0.281603551	0.182603415
0.458016778	0.169156519
0.617876244	0.149595989
0.755404408	0.124628971
0.865631202	0.095158512
0.944575023	0.062253524
0.989400935	0.027152459

表２－１．積分区間(-1, 1)におけるルジャンドル・ガウス積分公式の分点と重み係数の一部。

＜課題２のための準備＞

表２－１に掲げた係数をいちいち手で入力するのは大変面倒な作業であり、ミスを犯す可能性もある。そこで、あらかじめ用意されたデータファイルから係数を読み込むことにする。以下の手順にしたがってファイル入力用のプログラムを作成せよ。

1)「Mains」フォーム上で右クリックし、「メニューエディタ」を選択・クリックする。

2) 現れた入力フォームに図２－２と同じようにタイプし、OK を押す。そうすると、「File」というメニューができあがる。やり方がわからなければ TA に質問すること。

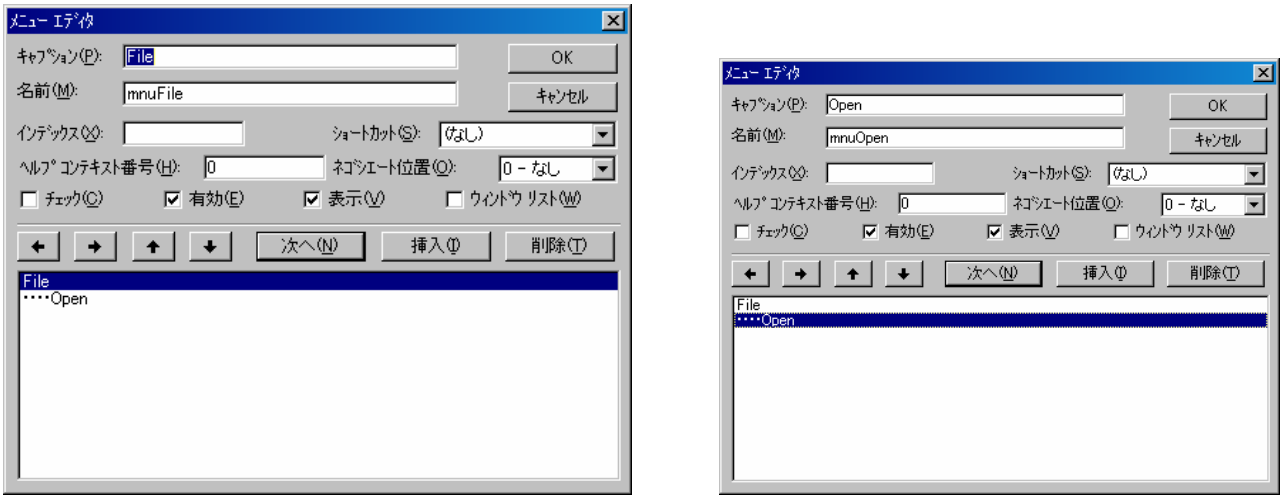

図２－２．メニューエディターの利用。

3) 新しく作った「File」メニューの中の「Open」をクリックして、プログラム入力画面を開く。そこに、以下のプログラムをそのまま入力する（図２－４も参照)。あるいは「ルジャンドル」という名前のフォルダ内に用意されたテキストファイル「open.txt」を「メモ帳」で開き、コピー&貼付けをしてもよい。

```
Private Sub mnuOpen_Click()
'-----データファイルのオープン-----
On Error GoTo errhandler
CommonDialog1.Filter = "すべてのファイル(*.*)|*.*|"
CommonDialog1.FilterIndex = 1
CommonDialog1.ShowOpen
'-----------------------------------------------

Open CommonDialog1.FileName For Input As #1  読込みモードで指定ファイルを#1 として開く。

'データファイルの読込み
Input #1, n       '分割点数を変数 n に入力
For i = 1 To n
   DoEvents
   For j = 1 To 2
      Input #1, dammy  'データファイルの係数を変数 dammy に入力
      w(i, j) = dammy     配列変数 w(i, j)にデータを入力
   Next j
Next i
Close #1

'読み込んだデータの確認
Mains.Print "n="; n
For i = 1 To n
    Mains.Print w(i, 1), w(i, 2)    'Mains に表示
Next i

MsgBox ("データ読込み完了")  'メッセージボックス
Exit Sub

'ファイル読込み中にエラーが起きたときの処理
errhandler:
    MsgBox ("エラー") 'メッセージボックス
    Close #1
End Sub
```

4) 変数の共通使用

手順 3)のプログラムで利用している配列変数 w(i, j)や多項式の次数 n のように、変数を幾つかのコマンドボタンやフォームにまたがって共通に使いたい場合、Mains フォーム内で共通変数として宣言しなければならない。やり方は次の通り。Mains フォームをダブルクリックし、左上に現れたプルダウンメニュー “ [Form　　|▾] ” の▾矢印を使って「(General)」に切り変え、また右の枠で “ [　　|▾] ” 内の「(Declarations)」を選び、次の Dim 文を書く(図２－３参照)。

```
Dim n%, w#(20, 20)
```

ここで宣言した変数を別のコマンドボタンの中で改めて宣言してしまうと、それらの変数はそのコマンドボタンの中だけで有効な変数となってしまい、共通性を失うので注意すること。

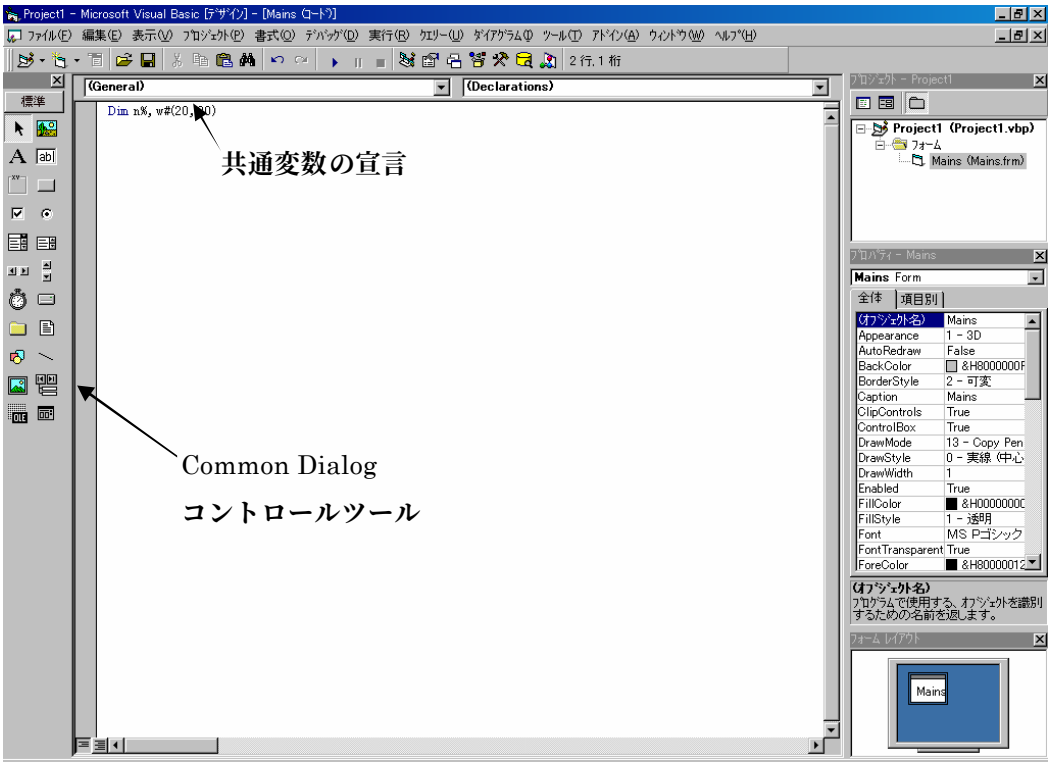

図２－３．共通変数の宣言の画面。Mains の「General」、「Declearation」**に宣言する。**

5）**メニュー→「プロジェクト(P)」→「コンポーネント」を選択し、**「Microsoft Common Dialog Control 6.0」**の左側にチェックを入れる。そして、ツールボックスから**「Common Dialog」**を選択し、フォーム上に貼り付ける（図２－３、図２－４）。**

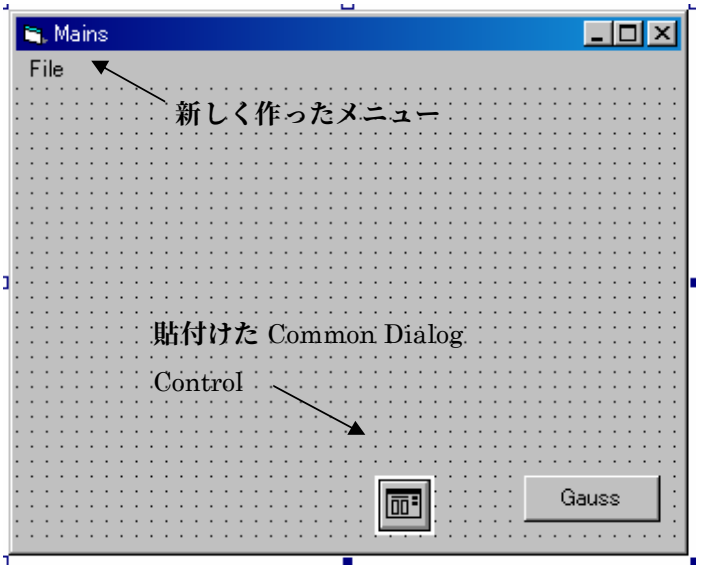

図２－４． Common Dialog Control **の利用。**

6）**プログラムを実行して、**「File」**メニューから**「Open」**を選び、あらかじめ用意してある「ルジャンドル」フォルダから係数のデータファイル（例えば** n16.txt）**を選んで係数が表示されることを確認する。**

以上で課題２の準備ができた。

(課題２)コマンドボタン Gauss を用意し、ルジャンドル・ガウスの積分公式(2-11)に従って

$$S=\int_0^1 \frac{\sqrt{x}}{x+2}\,dx$$

を計算するプログラムを作れ。分点 $n=3, 4, 8, 16$ の場合についてそれぞれ積分計算を実行し、結果をノートに記録する。

(ヒント) まず手計算によって変数変換$x=t^2$を施し、偶関数の性質を利用して積分を$\int_{-1}^{1} f(t)\,dt$ の形に直す。

３．モンテカルロ積分

図３－１に示したように、任意の長方形の内側に収まった関数 $f(x)$ がある。ここでは一様な乱数を使って関数 $f(x)$ の下側の面積を求める方法を考えよう。長方形の高さを h, 幅を $w=b-a$ とすると、長方形の面積は $h(b-a)$ である。この関数の下側の面積は次のようにして求められる。

①$a \le x \le b$, $0 \le y \le h$ を満たす一様な乱数の組(x, y)をn個発生させる。
②(x_i, y_i)が$y_i \le f(x_i)$を満たす乱数の組の個数n_Sを求める。
③乱数は一様であるから、関数の下側の面積の近似値は

$$S_n=\frac{n_S}{n}h(b-a) \qquad (3\text{-}1)$$

によって求められる。

以上の手続きから明らかなように、生成する乱数の個数を増やせば精度が上がると予想される。このような乱数を使った積分法をモンテカルロ積分という。

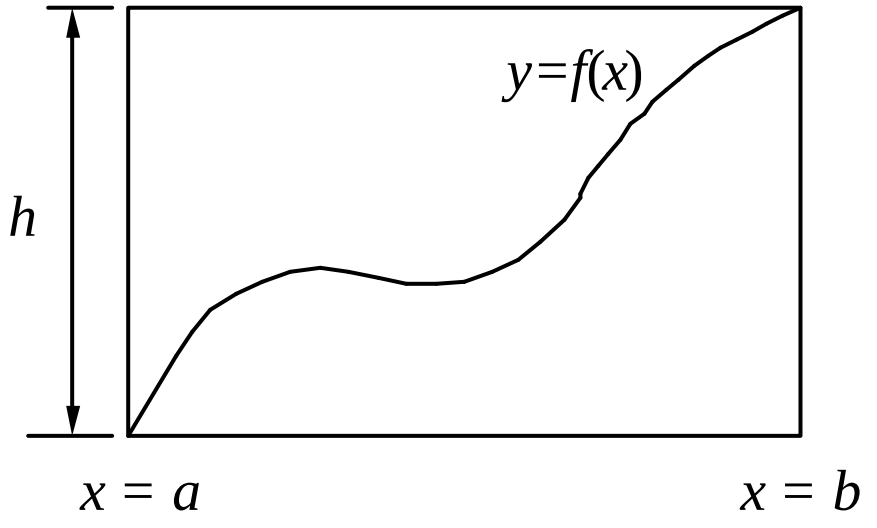

図３－１． 1次元のモンテカルロ積分

モンテカルロ積分には微積分学の「平均値の定理」を用いる方法もある。これは次のような方法で面積を求める。

①$a \le x \le b$を満たす一様な乱数x_iを生成し、関数の標本$f(x_i)$をランダムに多数“選び出す”。
②面積は標本$f(x_i)$の平均値から計算され

$$S_n = (b-a)\frac{1}{n}\sum_{i=1}^{n} f(x_i) \qquad (3\text{-}2)$$

で与えられる。

この式は長方形近似の(1-2)式と似ている。異なる点は和をとる間隔が乱数によってランダムにとられていることである。乱数は一様だから、nを増やしていくことによって(1-2)式と同様に計算精度が高くなる。

（課題３－１）コマンドボタン MC1（MC = MonteCarlo の略）を用意する。(3-1)式または(3-2)どちらかの式を用いて、モンテカルロ法により定積分(1-7)を求め、結果をフォームに印刷するプログラムを作成せよ。ただし$n = 10^4$とする。このとき得られた積分結果をノートに記録しておくこと。

（ヒント）疑似一様乱数を生成するコマンド Rnd( )を用いて区間(a, b)の一様乱数を求めるには次のようにすればよい（ヘルプで「Rnd( )」,「Randomize」を調べよ）。

```
a=1: b=5
x = (b-a)*Rnd( )+a
```

＜モンテカルロ法の誤差＞

モンテカルロ積分は計算のたびに異なる結果を与える（ただし、毎回同じ疑一様乱数列を用いた場合は同じ結果を与えることに注意）。計算結果の誤差の尺度は、多数回にわたるモンテカルロ積分の結果の**平均値の標準偏差**s_mによって与えられる。s_mは被積分関数の標本$f(x_i)$の標準偏差sから

$$s_m = \frac{s}{\sqrt{n-1}} \cong \frac{s}{\sqrt{n}}, \qquad (3\text{-}3)$$

と求められる。ただし

$$s = \sqrt{\frac{1}{n}\sum\left[f(x_i)^2\right] - (m)^2} \quad \text{（標本標準偏差）}, \quad m = \frac{1}{n}\sum f(x_i) \quad \text{（標本平均）} \qquad (3\text{-}4)$$

である。モンテカルロ法では誤差が$n^{-1/2}$に比例して 0 に近づくことになる（付録も参照）。

（課題３－２）コマンドボタン MC2 を用意する。MC1 の内容を利用して（コピーして）、定積分(1-7)を平均値の定理による方法(3-2)式に従って、標準偏差s_mがおよそ10^{-3}より小さくなるまで計算するプログラムに改良せよ。試行回数の初期値は$n = 10^5$からはじめ、積分が収束するまで10^5回づつ加算していくこと。積分値、試行回数n、標準偏差s_mは必ずノートに記録しておくこと。

＜第２回レポート課題＞

①自分で作成したプログラムリストに詳しいコメントをつけよ。また、それぞれの課題で計算した積分結果を提出すること。

②今回学んだ数値積分法の特徴を述べよ。また、利用するときに気をつけなければならない点を説明せよ（参考書図書、付録を調べる）。

☆課題①については次週までに提出すること。

（付録 A）数値積分の誤差

数値積分の誤差評価を行うため、被積分関数がテーラー展開可能であり、積分が

$$\int_{x_i}^{x_{i+1}} f(x)\,dx = f(x_i)h + \frac{1}{2}f'(x_i)h^2 + \frac{1}{6}f''(x_i)h^3 + \cdots, \quad h = x_{i+1} - x_i \tag{A-1}$$

と求められると仮定する。各区間の誤差をd_iとすると、長方形近似による積分の誤差は

$$d_i = \int_{x_i}^{x_{i+1}} f(x)\,dx - f(x_i)h \approx \frac{1}{2}f'(x_i)h^2 \tag{A-2}$$

となる。すなわち各区間での積分の主要な誤差はh^2であることが分かる。全体では区間数がnであるから誤差dは

$$d \approx nh^2 = n \times \frac{(b-a)^2}{n^2} \propto n^{-1} \tag{A-3}$$

となり、n^{-1}に比例する。同様に、台形近似の場合には

$$d_i = \int_{x_i}^{x_{i+1}} f(x)\,dx - \frac{h}{2}\left[f(x_i) + f(x_{i+1})\right] \tag{A-4}$$

ここで、$f(x)$に対してx_iのまわりでテーラー展開を行うと

$$f(x) = f(x_i) + f'(x_i)(x - x_i) + \frac{1}{2}f''(x_i)(x - x_i)^2 + \frac{1}{6}f'''(x_i)(x - x_i)^3 + \cdots \tag{A-5}$$

であるから、$x \to x_{i+1}$として

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{1}{2}f''(x_i)h^2 + \frac{1}{6}f''(x_i)h^3 + \cdots \tag{A-6}$$

となる。(A-1)と(A-6)を(A-4)に代入すると

$$d_i \approx f(x_i)h + \frac{1}{2}f'(x_i)h^2 + \frac{1}{6}f''(x_i)h^3 - \frac{h}{2}\left[f(x_i) + f(x_i) + f'(x_i)h + \frac{1}{2}f''(x_i)h^2 + \frac{1}{6}f'''(x_i)h^3\right]$$

$$= \frac{h^3}{12}f'''(x_i)$$

従って、全体の誤差は

$$d \approx n \times \frac{h^3}{12}f'''(x_i) \propto \frac{1}{n^2} \tag{A-7}$$

となり、n^{-2}に比例する。同様にシンプソン法では$f(x)$を区間ごとに 2 次曲線で近似するので、各区間の誤差がh^5に依存していることがわかる。従って、全体の誤差はn^{-4}に比例する。

２次元積分の誤差の評価に対しても同様の議論を行う。$z = f(x, y)$は曲面を表している。この曲面の下側の体積が２次元積分の値である。曲面下の体積を求めるときに「小さな直方体$h_x h_y$」の和として求める近似を考える。誤差を求めるために$z = f(x, y)$をテーラー展開すると

$$f(x, y) = f(x_i, y_i) + \frac{\partial f(x_i, y_i)}{\partial x}(x - x_i) + \frac{\partial f(x_i, y_i)}{\partial y}(y - y_i) + \cdots \tag{A-8}$$

となる。1 次元の場合と同様に考えると積分の誤差は

$$d_i = \iint f(x, y)\,dxdy - f(x_i, y_i)h_x h_y \tag{A-9}$$

とかける。直方体近似では

$$\iint f(x,y)\,dxdy \approx f(x_i,y_i)h_x h_y + \frac{f'(x_i,y_i)}{2}h_x^2 h_y + \frac{f'(x_i,y_i)}{2}h_y^2 h_x \qquad \text{(A-10)}$$

であり、h_x と h_y は同じ程度の大きさ h であるとすれば

$$d_i \approx \frac{h^3}{2} f'(x_i,y_i) \qquad \text{(A-11)}$$

となる。全体積 V を直方体で n 分割したとすると、２次元では $n = V/h^2$ であるから、全体の誤差は

$$d \propto n \times h^3 \propto \frac{1}{h^2} \times h^3 = h = \frac{1}{n^{1/2}} \qquad \text{(A-12)}$$

となり、$n^{-1/2}$ に比例する。台形近似、シンプソン法でもまったく同じであり、一般に「1 **次元での誤差の次数が** n^{-a} **なら、** d **次元積分の誤差の次数は** $n^{-a/d}$ 」に比例するようになる。つまり、高次元になれば誤差の減少が緩くなり、積分の収束性も悪くなる。一方、モンテカルロ積分では試行回数を n とすると次元によらず全体の誤差は $n^{-1/2}$ の依存性をもつ。よって高次元積分ではモンテカルロ法が有利になる。

（オプション課題１）コマンドボタン MC3 を用意する。定積分(1-7)を平均値の定理による方法(3-2)式を用いて $n = 10^4$ **として 100 個の積分値を求め、平均値と標準偏差を求めるプログラムを作成せよ。その結果を(3-4)式の結果と比較し、結果がそれほど変わらないことを確認せよ。**

（オプション課題２）半径 r **の球の体積をモンテカルロ法によって求めるプログラムを作成し、** $4\pi r^3/3$ **と比較せよ。また一般に** n **次元の超球体**

$x_1^2 + x_2^2 + \cdots + x_n^2 \le r^n$

の体積を求めるプログラムを作成せよ（ $r = 1$ **で作ってみよ）。**